

CLAIMS

1. A computer-based method for automatically finding and exploiting hidden, fuzzy algebraic constraints in a database, said method comprising the steps of:

(a) constructing one or more candidates of form $C=(a_1, a_2, P, \oplus)$, wherein a_1 and
5 a_2 are numerical attributes associated with column values of data in said database, P is a pairing rule, and \oplus is any of the following algebraic operators: +, -, \times , or /;

(b) constructing, for each candidate identified in (a), an algebraic constraint $AC=(a_1, a_2, P, \oplus, I_1, \dots, I_k)$ by applying any of, or a combination of the following techniques to a sample of column values: statistical histogramming, segmentation, or
10 clustering, where I_1, \dots, I_k is a set of disjoint intervals and $k \geq 1$, and

wherein said constructed algebraic constraints are used in query optimization.

2. A compute-based method as per claim 1, wherein one or more pruning rules are used to limit said number of constructed candidates.

15

3. A computer-based method as per claim 2, wherein said pairing rule P represents either a trivial pairing rule \emptyset_R or a join between tables R and S and said pruning rules comprise any of, or a combination of the following:

pairing rule P is of form $R.a = S.b$ or of the form \emptyset_R , and the number of rows in
20 either table R or table S lies below a specified threshold value;

pairing rule P is of form $R.a = S.b$ with $a \in K$ and the number of distinct values in $S.b$ divided by the number of values in $R.a$ lies below a specified threshold value, wherein K is a set comprising key-like columns among all columns in said database;

pairing rule P is of form $R.a = S.b$, and one or both of R and S fails to have an
5 index on any of its columns; or

pairing rule P is of form $R.a = S.b$ with $a \in K$, and $S.b$ is a system-generated key.

4. A computer-based method as per claim 1, wherein said method further comprises the steps of:

10 identifying a set of useful algebraic constraints via one or more pruning rules; and
partitioning data into compliant data and exception data.

5. A computer-based method as per claim 4, wherein said method further comprises the steps of:

15 receiving a query;
modifying said query to incorporate identified constraints; and
combining results of modified query executed on data in said database and said
original query executed on exception data.

6. A computer-based method as per claim 4, wherein said partitioning is done by incrementally maintained materialized views, partial indices, or physical partitioning of the table.

5 7. A computer-based method as per claim 2, wherein said pruning rules comprise any of, or a combination of the following:

a_1 and a_2 are not comparable data types;

the fraction of NULL values in either a_1 or a_2 exceeds a specified threshold; or

either column a_1 or a_2 is not indexed.

10

8. A computer-based method as per claim 1, wherein said step of constructing one or more candidates further comprises the steps of:

generating a set \mathcal{P} of pairing rules; and

for each pairing rule $P \in \mathcal{P}$, systematically considering possible attribute pairs $(a_1,$

15 $a_2)$ and operators \oplus with which to construct candidates.

9. A computer-based method as per claim 8, wherein said step of generating a set \mathcal{P} of pairing rules further comprises the steps of:

initializing \mathcal{P} to be an empty set;

20 adding a trivial pairing rule of the form \emptyset_R to said set \mathcal{P} for each table R in said

database; and

generating and adding nontrivial pairing rules to said set \mathcal{P} based upon identifying matching columns via an inclusion dependency, wherein a column b is considered a match for column a if:

data in columns a and b are of a comparable type; or

- 5 either (i) column a is a declared primary key and column b is a declared foreign key for the primary key, or (ii) every data value in a sample from column b has a matching value in column a .

- 10 10. A computer-based method as per claim 8, wherein said step of generating a set \mathcal{P} of pairing rules further comprises the steps of:

initializing \mathcal{P} to be an empty set;

adding a trivial pairing rule of the form \emptyset_R to said set \mathcal{P} for each table R in said database; and

- 15 generating a set K of key-like columns from among all columns in said database with each column in set K belonging to a predefined set of types T , said set K comprising declared primary key columns, declared unique key columns, and undeclared key columns, wherein said primary keys or declared unique keys are compound keys of form $a = (a_1, \dots, a_m) \in T^m$ for $m > 1$;

- 20 adding nontrivial pairing rules to said set \mathcal{P} based upon identifying matching compound columns via an inclusion dependency wherein, given a compound key

$(a_1, \dots, a_m) \in K$, a compound column b is considered a component wise match for compound column a if:

data in compound columns a and b are of a comparable type; or

either (i) compound column a is a declared primary key and compound
 5 column b is a declared foreign key for the primary key, or (ii) every data value in a
 sample from compound column b has a matching value in compound column a .

11. A computer-based method as per claim 1, wherein said step of constructing algebraic constraints further comprises the steps of:

10 constructing a sample set W_C of an induced set Ω_C , wherein P is a join predicate between tables R and S and $\Omega_C = \{r.a_1 \oplus r.a_2 : r \in R\}$ when the pairing rule P is a trivial rule \emptyset_R and $\Omega_C = \{r.a_1 \oplus s.a_2 : r \in R, s \in S, \text{ and } (r, s) \text{ satisfies } P\}$;

sorting n data points in said sampled set W_C in increasing order as $x_1 \leq x_2 \leq \dots \leq x_n$
 and constructing a set of disjoint intervals I_1, \dots, I_k such that data in sample W_C falls
 15 within one of said disjoint intervals, wherein segmentation for constructing said set of
 disjoint intervals is specified via a vector of indices $(i(1), i(2), \dots, i(k))$ and the j^{th} interval
 is given by $I_j = [x_{i(j-1)+1}, x_{i(j)}]$ and length of I_j , denoted by L_j , is given by
 $L_j = x_{i(j)} - x_{i(j-1)+1}$; and

wherein the function for optimizing cost associated with said segmentation is

20 $c(S) = wk + (1 - w) \left[\frac{1}{\Delta} \sum_{j=1}^k L_j \right]$ with w being a fixed weight between 0 and 1 and a

segmentation that minimizes c is defined by placing adjacent points x_l and x_{l+1} in the same segment if and only if $x_{l+1} - x_l < d^*$, where $d^* = \Delta(w/(1-w))$.

12. A computer-based method as per claim 11, wherein widths associated with said
5 intervals are expanded to avoid additional sampling required to increase right end point to equal maximum value in Ω_C .

13. A computer-based method as per claim 11, wherein size of said sampled set is approximated via the following iterative steps:

10 (a) given a k -segmentation, setting counters $i=1$ and $k=1$;

(b) selecting a sample size $n=n^*$, wherein $n^*(k) \approx \frac{\chi^2_{1-p}(2-f)}{4f} + \frac{k}{2}$, wherein p is

the probability that at least a fraction of points in Ω_C that lie outside the intervals is at most f ;

(c) obtaining a sample based on (b), computing algebraic constraints, and
15 identifying a number k' of bump intervals; and

(d) if $n \geq n^*(k')$ or $i = i_{max}$, then utilizing sample size in (b); else setting counters $k=k'$ and $i=i+1$, and returning to step (b).

14. A computer-based method as per claim 11, wherein $\Delta = x_n - x_1$.

20

15. A computer-based method as per claim 11, wherein Δ is estimated as a generic function $f(a_1^m, a_1^M, a_2^m, a_2^M)$, wherein a_1^M and a_1^m are maximum and minimum values associated with column a_1 , and a_2^M and a_2^m are maximum and minimum values associated with column a_2 .

5

16. A computer-based method as per claim 1, wherein said method is implemented across networks.

17. A computer-based method as per claim 16, wherein said across networks element
10 comprises any of, or a combination of the following: local area network (LAN), wide area network (WAN), or the Internet.

18. A computer-based method as per claim 1, wherein said step of constructing algebraic constraints further comprises the steps of:

15 constructing a sample set W_C of an induced set Ω_C , wherein P is a join predicate between tables R and S and $\Omega_C = \{r.a_1 \oplus r.a_2 : r \in R\}$ when the pairing rule P is a trivial rule \emptyset_R and $\Omega_C = \{r.a_1 \oplus s.a_2 : r \in R, s \in S, \text{ and } (r, s) \text{ satisfies } P\}$;

sorting n data points in said sampled set W_C in increasing order as $x_1 \leq x_2 \leq \dots \leq x_n$

and constructing a set of disjoint intervals I_1, \dots, I_k such that data in sample W_C falls

20 within one of said disjoint intervals, wherein segmentation for constructing said set of

disjoint intervals is specified via a vector of indices $(i(1), i(2), \dots, i(k))$ and the j^{th} interval

is given by $I_j=[x_{i(j-1)+1}, x_{i(j)}]$ and length of I_j , denoted by L_j , is given by $L_j = x_{i(j)} - x_{i(j-1)+1}$; and said segments are determined by histogramming.

19. A computer-based method as per claim 18, wherein said histogramming is done
5 using $2h(n)$ buckets when $h(n)=(2n)^{1/3}$ is the oversmoothing lower bound.

20. A computer-based method as per claim 18, wherein widths associated with said intervals are expanded to avoid additional sampling required to increase right end point to equal maximum value in Ω_C .

10

21. A computer-based method as per claim 18, wherein size of said sampled set is approximated via the following iterative steps:

(a) given a k -segmentation, setting counters $i=1$ and $k=1$;

(b) selecting a sample size $n=n^*$, wherein $n^*(k) \approx \frac{\chi^2_{1-p}(2-f)}{4f} + \frac{k}{2}$, wherein p is

15 the probability that at least a fraction of points in Ω_C that lie outside the intervals is at most f ;

(c) obtaining a sample based on (b), computing algebraic constraints, and identifying a number k' of bump intervals; and

(d) if $n \geq n^*(k')$ or $i = i_{max}$, then utilizing sample size in (b); else setting counters
20 $k=k'$ and $i=i+1$, and returning to step (b).

22. An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements a method for automatically finding and exploiting hidden, fuzzy algebraic constraints in a database,
5 said method comprising the steps of:

- (a) computer readable program code constructing one or more candidates of form $C=(a_1, a_2, P, \oplus)$, wherein a_1 and a_2 are numerical attributes associated with column values of data in said database, P is a pairing rule, and \oplus is any of the following algebraic operators: +, -, \times , or /;
- 10 (b) computer readable program code constructing, for each candidate identified in (a), an algebraic constraint $AC=(a_1, a_2, P, \oplus, I_1, \dots, I_k)$ by applying any of, or a combination of the following techniques to a sample of column values: statistical histogramming, segmentation, or clustering, where I_1, \dots, I_k is a set of disjoint intervals and $k \geq 1$, and
- 15 wherein said constructed algebraic constraints are used in query optimization.

23. An article of manufacture as per claim 22, wherein said medium further comprises:

- computer readable program code identifying a set of useful algebraic constraints
- 20 via heuristics comprising a set of pruning rules; and

computer readable program code partitioning data into compliant data and exception data.

24. An article of manufacture as per claim 23, wherein said medium further
5 comprises:

computer readable program code aiding in receiving a query;

computer readable program code modifying said query to incorporate identified constraints; and

computer readable program code combining results of modified query executed on
10 data in said database and said original query executed on exception data.

25. A computer-based method for optimizing a query executed on relational data in a database, said method comprising the steps of:

(a) constructing one or more candidates of form $C=(a_1, a_2, P, \oplus)$, wherein a_1 and
15 a_2 are numerical attributes associated with column values of data in said database, P is a pairing rule, and \oplus is any of the following algebraic operators: +, -, \times , or /;

(b) constructing, for each candidate identified in (a), a fuzzy algebraic constraint $AC=(a_1, a_2, P, \oplus, I_1, \dots, I_k)$ by applying any of, or a combination of the following techniques to a sample of column values: statistical histogramming, segmentation, or
20 clustering, where I_1, \dots, I_k is a set of disjoint intervals and $k \geq 1$;

(c) identifying a set of useful algebraic constraints via heuristics comprising a set of pruning rules;

(d) partitioning data into compliant data and exception data; and

(e) modifying said query to incorporate identified constraints, wherein an optimizer utilizes said identified constraints to identify new and efficient paths.

26. A computer-based method as per claim 25, wherein said method further comprises the step of combining results of modified query executed on data in said database and said original query executed on exception data.

10

27. A computer-based method as per claim 25, wherein said partitioning is done by incrementally maintained materialized views, partial indices, or physical partitioning of the table.

15 28. A computer-based method as per claim 25, wherein said pairing rule P represents either a trivial pairing rule \emptyset_R or a join between tables R and S and said pruning rules comprise any of, or a combination of the following:

pairing rule P is of form $R.a = S.b$ or of the form \emptyset_R , and the number of rows in either table R or table S lies below a specified threshold value;

pairing rule P is of form $R.a = S.b$ with $a \in K$ and the number of distinct values in $S.b$ divided by the number of values in $R.a$ lies below a specified threshold value, wherein K is a set comprising key-like columns among all columns in said database;

pairing rule P is of form $R.a = S.b$, and one or both of R and S fails to have an
5 index on any of its columns; or

pairing rule P is of form $R.a = S.b$ with $a \in K$, and $S.b$ is a system-generated key.

29. A computer-based method as per claim 25, wherein said step of constructing one or more candidates further comprises the steps of:

10 generating a set \mathcal{P} of pairing rules; and

for each pairing rule $P \in \mathcal{P}$, systematically considering possible attribute pairs (a_1, a_2) and operators \oplus with which to construct candidates.

30. A computer-based method as per claim 29, wherein said step of generating a set \mathcal{P}
15 of pairing rules further comprises the steps of:

initializing \mathcal{P} to be an empty set;

adding a trivial pairing rule of the form \emptyset_R to said set \mathcal{P} for each table R in said
database; and

generating and adding nontrivial pairing rules to said set \mathcal{P} based upon identifying
20 matching columns via an inclusion dependency, wherein a column b is considered a

match for column a if:

ARC920030044US1

data in columns a and b are of a comparable type; or

either (i) column a is a declared primary key and column b is a declared foreign key for the primary key, or (ii) every data value in a sample from column b has a matching value in column a .

5

31. A computer-based method as per claim 29, wherein said step of generating a set \mathcal{P} of pairing rules further comprises the steps of:

initializing \mathcal{P} to be an empty set;

adding a trivial pairing rule of the form \emptyset_R to said set \mathcal{P} for each table R in said

10 database; and

generating a set K of key-like columns from among all columns in said database with each column in set K belonging to a predefined set of types T , said set K comprising declared primary key columns, declared unique key columns, and undeclared key columns, wherein said primary keys or declared unique keys are compound keys of form

15 $a = (a_1, \dots, a_m) \in T^m$ for $m > 1$;

adding nontrivial pairing rules to said set \mathcal{P} based upon identifying matching compound columns via an inclusion dependency wherein, given a compound key $(a_1, \dots, a_m) \in K$, a compound column b is considered a component wise match for compound column a if:

20 data in compound columns a and b are of a comparable type; or

either (i) compound column a is a declared primary key and compound column b is a declared foreign key for the primary key, or (ii) every data value in a sample from compound column b has a matching value in compound column a .

5 32. A computer-based method as per claim 25, wherein said method is implemented across networks.

33. A computer-based method as per claim 32, wherein said across networks element comprises any of, or a combination of the following: local area network (LAN), wide
10 area network (WAN), or the Internet.

34. A method to optimize query processing in a database comprising the steps of:

(a) identifying candidates of form $C = (a_1, a_2, P, \oplus)$ by finding declared or undeclared key columns and columns related to said declared and undeclared key
15 columns via an inclusion dependency, wherein a_1 and a_2 are numerical attributes associated with column values of data in said database, P is a pairing rule, and \oplus is an algebraic operator;

(b) for each candidate in (a), identifying a sample set and constructing an algebraic constraint $AC = (a_1, a_2, P, \oplus, I_1, \dots, I_k)$ for said sample set by applying
20 any of, or a combination of the following techniques: statistical histogramming, segmentation, or clustering techniques, wherein the sample size is selected to

control the number of exception records that fail to satisfy said algebraic constraint;

(c) identifying a set of useful constraints and associated exception tables via heuristics comprising pruning rules and creating exception tables to hold said exception records; and

(d) during query processing, modifying queries to incorporate said identified algebraic constraints with an optimizer utilizing said identified algebraic constraints and said created exception tables to accelerate query processing.

35. A query optimization method based upon discovering undeclared and fuzzy rules in a database, said method comprising the steps of:

(a) identifying a set of keys for one or more tables in said database;

(b) identifying a set of inclusion dependencies covered by said identified keys;

(c) identifying sets of column pairs in said tables linked by an inclusion dependency,

(d) sampling data from columns in each column pair identified in (c), and

(e) using a data mining algorithm to identify significant patterns between data in said columns and utilizing said patterns to derive one or more rules, wherein query optimization is attained by modifying a query to incorporate constraints defined by said derived one or more rules.

36. A query optimization method as per claim 35, wherein said method further comprises the step of partitioning data into compliant data and exception data based upon said identified patterns and said derived one or more rules.

5 37. A query optimization method as per claim 36, wherein said partitioning is done by incrementally maintained materialized views, partial indices, or physical partitioning of the table.

38. A query optimization method as per claim 36, wherein said step of modifying
10 query to incorporate constraints further comprises the steps of:

receiving a query;

modifying said query to incorporate identified constraints; and

combining results of modified query executed on data in said database and said
original query executed on exception data.

15